

Proof Theory, Semantics and Algebra for Normative Systems

Xin Sun

Individual and Collective Reasoning Group, University of Luxembourg
xin.sun@uni.lu

Abstract. This paper reports a correspondence between input/output logic and the theory of joining-system, an algebraic approach to normative system. The results have the form: every norm (a, x) is logically derivable from a set of norms G if and only if it is in the space of norms algebraically generated by G . We present three versions of correspondence: input/output logic and Boolean joining-system, intuitionistic input/output logic and Heyting joining-system, quasi input/output logic and quasi joining-system. The algebraic approach offers a holistic perspective on normative systems. We use isomorphism and embedding of joining-system to discuss the similarity of normative systems.

Key words: normative system, input/output logic, joining-system, deontic logic

1 Introduction

In their influential book *Normative Systems* [1], Alchourrón and Bulygin conceive a normative system as a deductive mechanism, like a black box which produces normative statements as output, when we feed it descriptive statements as input. To this tradition also belong various input/output logics [10–12, 18, 20] and the theory of joining-system [8, 9].

Although sharing the same ancestor, input/output logic and joining-system have evolved quite separately, and look very different at first sight. Input/output logic has proof systems and semantics, while joining-system uses algebra as a tool to model normative systems without referring to proof systems or semantics. In this paper, we show that the two accounts essentially give the same results, and can be seen as “two sides of one and the same coin.” The results illustrate that proof theory, semantics and algebra, as three tools to model normative systems, provide different perspectives on the investigation of normative systems. For example, the algebraic approach gives rise to holistic results to normative systems in the sense that we can use algebraic concepts such as embedding and isomorphism to compare the structural similarity of different normative systems. It is the different features of these three tools that motivate us to use all of them.

The layout of this paper is as follows. In Section 2 we give a brief introduction to input/output logic and joining-system. Then, from Section 3 to 5 we present

correspondence results between input/output logic and joining-system. Section 6 is the section for applications of joining-system. We use examples to illustrate the holistic views we gain by the algebraic representation of normative systems. In Section 7 we present some issues for future research and summarize this paper. For the sake of readability, all proofs are listed in the appendix.

2 Background

2.1 Input/output logic

In a series of papers [11–13], Makinson and van der Torre develop a class of deontic logic called input/output logic. A gentle and comprehensive introduction can be found in Makinson and van der Torre [14] and Parent and van der Torre [19] respectively. In general, the matured version of input/output logic is the constrained version [12]. For simplicity's sake, the latter is put aside, and only unconstrained input/output logics are considered in this paper.

Let $P = \{p_0, p_1, \dots\}$ be a countable set of propositional letters and \mathbb{L}_P be the propositional language built upon P together with logical constants \top and \perp . Let G be a set of ordered pairs of formulas of \mathbb{L}_P . A pair $(a, x) \in G$, call it a norm, is read as “given a , it ought to be x ”. A normative system is a set of norms. G is viewed as a function from $2^{\mathbb{L}_P}$ to $2^{\mathbb{L}_P}$ such that for a set A of formulas, $G(A) = \{x : (a, x) \in G \text{ for some } a \in A\}$.

Makinson and van der Torre define out_1 to out_4 as following:

- $out_1(G, A) = Cn(G(Cn(A)))$
- $out_2(G, A) = \bigcap \{Cn(G(V)) : A \subseteq V, V \text{ is complete}\}$
- $out_3(G, A) = \bigcap \{Cn(G(B)) : A \cup G(B) \subseteq B = Cn(B)\}$
- $out_4(G, A) = \bigcap \{Cn(G(V)) : A \cup G(V) \subseteq V, V \text{ is complete}\}$

Here Cn is the classical consequence operator of propositional logic, and a complete set is a set of formulas that is either maxi-consistent or equal to \mathbb{L}_P . $out_1(G, A)$ to $out_4(G, A)$ are called *simple-minded output*, *basic output*, *simple-minded reusable output* and *basic reusable output* respectively.

The proof system of input/output logic is based on derivation of ordered pairs of formulas. An ordered pair of formulas is derivable from a set G iff (a, x) is in the least set that includes G and closed under a number of rules. The following are the rules used in input/output logic:

- SI (strengthening the input): from (a, x) to (b, x) whenever $b \vdash a$
- WO (weakening the output): from (a, x) to (a, y) whenever $x \vdash y$
- AND (conjunction of output): from $(a, x), (a, y)$ to $(a, x \wedge y)$
- OR (disjunction of input): from $(a, x), (b, x)$ to $(a \vee b, x)$
- CT (cumulative transitivity): from $(a, x), (a \wedge x, y)$ to (a, y)

The derivation system based on the rules SI, WO and AND is called $deriv_1$. Adding OR and CT to $deriv_1$ gives $deriv_2$ and $deriv_3$ respectively. The five rules together define $deriv_4$. We use $(a, x) \in deriv_i(G)$, or equivalently $x \in$

$deriv_i(G, a)$, to denote the norm (a, x) is derivable from G using rules of derivation system $deriv_i$. Moreover, for a set A of formulas, we use $x \in deriv_i(G, A)$ to denote the fact that there exist $a_1 \dots a_n \in A$ such that $(a_1 \wedge \dots \wedge a_n, x) \in deriv_i(G)$. In Makinson and van der Torre [11], the following completeness theorems are given:

Theorem 1 ([11]). *Given an arbitrary normative system G and a set A of formulas, for $i \in \{1, 2, 3, 4\}$*

- $x \in out_i(G, A)$ iff $x \in deriv_i(G, A)$

2.2 Theory of joining-system

An algebraic framework for analyzing normative systems was introduced by Lars Lindahl and Jan Odelstad [7, 16, 8, 9]. The most general form of the theory is called theory of joining-system [9]. A joining-system is a triple (B_1, B_2, S) where B_1, B_2 are two ordered algebraic structures and S a relation between B_1 and B_2 satisfying some conditions. In Lindahl and Odelstad's work, the algebraic structure is usually a Boolean quasi-ordering. In this section we discuss Boolean algebra and move to Heyting algebra and Boolean quasi-ordering in Section 4 and 5 respectively.

Definition 1 (Boolean algebra). *A structure $\mathfrak{A} = (A, +, \cdot, -, 0, 1)$ is a Boolean algebra iff it satisfies the following identities:*

1. $x + y = y + x, \quad x \cdot y = y \cdot x$
2. $x + (y + z) = (x + y) + z, \quad x \cdot (y \cdot z) = (x \cdot y) \cdot z$
3. $x + 0 = x, \quad x \cdot 1 = x$
4. $x + (-x) = 1, \quad x \cdot (-x) = 0$
5. $x + (y \cdot z) = (x + y) \cdot (x + z), \quad x \cdot (y + z) = (x \cdot y) + (x \cdot z)$

The elements of a Boolean algebra are ordered as $a \leq b$ iff $a \cdot b = a$. It can be proved that \leq is reflexive, transitive, and anti-symmetric, therefore \leq is a partial order.

Given two Boolean algebras $\mathfrak{A} = (A, +_A, \cdot_A, -_A, 0_A, 1_A)$ and $\mathfrak{B} = (B, +_B, \cdot_B, -_B, 0_B, 1_B)$ ¹ with ordering \leq_A and \leq_B respectively. For two ordered pairs $(a, x), (b, y) \in A \times B$, we define $(a, x) \preceq (b, y)$ iff $b \leq_A a$ and $x \leq_B y$. We say (a, x) is narrower than (b, y) if $(a, x) \preceq (b, y)$.²

Lindahl and Odelstad use their narrowness relation to define joining-system to algebraically represent normative systems. To build the correspondence between input/output logic and joining-system, we introduce joining-system which are variants of Lindahl and Odelstad's.

¹ For the sake of simplicity, in the rest of this paper a Boolean algebra \mathfrak{A} always has the structure $(A, +_A, \cdot_A, -_A, 0_A, 1_A)$, and similarly for a Boolean algebra \mathfrak{B} .

² Such a narrowness relation is the same as the subinterval relation of Odelstad and Boman [15].

We define those variants based on Boolean algebra. Resembling the names of input/output logic, we call those variants of joining-system: *simple-minded*, *basic*, *simple-minded reusable* and *basic reusable* respectively. The basic Boolean joining-system is the one most similar to Lindahl and Odelstad's. The other three variants are introduced in the next section.

Definition 2 (Basic Boolean joining-system). *A basic Boolean joining-system is a structure $\mathbb{S} = (\mathfrak{A}, \mathfrak{B}, S)$ such that \mathfrak{A} and \mathfrak{B} are Boolean algebras and $S \subseteq A \times B$ satisfies the following conditions:*

1. *If $(a, x) \in S$ and $(a, x) \preceq (b, y)$, then $(b, y) \in S$.*
2. *For all finite $X \subseteq B$, if for all $x \in X, (a, x) \in S$, then $(a, y) \in S$ for all $y \in glb(X)$.³*
3. *For all finite $X \subseteq A$, if for all $x \in X, (x, b) \in S$, then $(y, b) \in S$ for all $y \in lub(X)$.⁴*

If $\mathbb{S} = (\mathfrak{A}, \mathfrak{B}, S)$ is a basic Boolean joining-system, then we call S a basic Boolean joining space. We equivalently replace condition 2 and 3 by the following respectively, and use them in later proofs:

- 2' *If $(a, x) \in S$ and $(a, y) \in S$, then $(a, x \cdot_B y) \in S$*
- 3' *If $(a, x) \in S$ and $(b, x) \in S$, then $(a +_A b, x) \in S$*

Moreover, we can equivalently define basic Boolean joining space using the standard algebraic terminology of ideal and filter:

Definition 3 (Ideal). *Let \mathfrak{A} be a Boolean algebra and I a subset of A . For I to be an ideal of \mathfrak{A} , it is necessary and sufficient that the following three conditions be satisfied:*

1. $0_A \in I$
2. *for all $x, y \in I$, $x +_A y \in I$*
3. *for all $x \in I$ and $y \in A$, if $y \leq_A x$ then $y \in I$*

Definition 4 (Filter). *Let \mathfrak{A} be a Boolean algebra and F a subset of A . For F to be a filter of \mathfrak{A} , it is necessary and sufficient that the following three conditions are satisfied:*

1. $1_A \in F$
2. *for all $x, y \in F$, $x \cdot_A y \in F$*
3. *for all $x \in F$ and $y \in A$, if $x \leq_A y$ then $y \in F$*

Let $F_{\uparrow}(X)$ be the filter generated by X and $I_{\downarrow}(X)$ be the ideal generated by X . Then we have the following proposition defining joining space by ideal and filter:

³ Here *glb* is the abbreviation of greatest lower bound. Formally, $glb(X) = \{b : \forall x \in X, b \leq x \text{ and } \forall a, \text{ if } \forall x \in X, a \leq x, \text{ then } a \leq b\}$

⁴ *lub* is the abbreviation of least upper bound. Formally, $lub(X) = \{a : \forall x \in X, x \leq a \text{ and } \forall b, \text{ if } \forall x \in X, x \leq b, \text{ then } a \leq b\}$

Proposition 1. *Given a structure $\mathbb{S} = (\mathfrak{A}, \mathfrak{B}, S)$, where $\mathfrak{A}, \mathfrak{B}$ are Boolean algebras and $S \subseteq A \times B$, S is a basic Boolean joining space in \mathbb{S} if and only if it satisfies the following conditions:*

1. *For every finite set $X \subseteq B$ and every $a \in A$, if for every $x \in X$, $(a, x) \in S$, then for every $y \in F_{\uparrow}(X)$, $(a, y) \in S$.*
2. *For every finite set $X \subseteq A$ and every $a \in A$, if for every $x \in X$, $(x, b) \in S$, then for every $y \in I_{\downarrow}(X)$, $(y, b) \in S$.*

Up to now, we have defined basic Boolean joining-system and joining space. But does a basic Boolean joining space always exist? The answer is positive. As the following proposition shows, the largest and the smallest basic Boolean joining space always exists.

Proposition 2. *Given two Boolean algebras $\mathfrak{A}, \mathfrak{B}$,*

1. *$(\mathfrak{A}, \mathfrak{B}, A \times B)$ is a basic Boolean joining-system.*
2. *If for all $i \in I$, $(\mathfrak{A}, \mathfrak{B}, S_i)$ is a basic Boolean joining-system, then $(\mathfrak{A}, \mathfrak{B}, \bigcap_{i \in I} S_i)$ is a basic Boolean joining-system.*

3 Input/output logic and joining-system

3.1 Basic input/output logic and joining-system

In this subsection, we prove that for a set of norms G , a norm (a, x) is derivable from G in basic input/output logic, if and only if it is in the basic Boolean joining space generated by G . To show this, we introduce a special Boolean algebra named Lindenbaum-Tarski algebra.

Let \equiv be the provable equivalence relation on \mathbb{L}_P , i.e. for every formula $\phi, \psi \in \mathbb{L}_P$, $\phi \equiv \psi$ iff $\vdash \phi \leftrightarrow \psi$. Let $\mathbb{L}_{\overline{P}}$ be the set of equivalence classes that \equiv induces on \mathbb{L}_P . For any formula $\phi \in \mathbb{L}_P$, let $[\phi]$ denote the equivalence class containing ϕ .

Definition 5 (Lindenbaum-Tarski algebra). *The Lindenbaum-Tarski algebra for propositional logic \mathbb{L}_P is a structure $\mathfrak{L} = (\mathbb{L}_{\overline{P}}, +, \cdot, -, 0, 1)$ where $[\phi] + [\psi] = [\phi \vee \psi]$, $[\phi] \cdot [\psi] = [\phi \wedge \psi]$, $-[\phi] = [\neg\phi]$, $0 = [\perp]$ and $1 = [\top]$.*

For more details of Lindenbaum-Tarski algebra, readers are suggested to consult Chapter 5 of Blackburn [4]. Every Lindenbaum-Tarski algebra is a Boolean algebra.

Let G be a set of norms. Let $G^{\equiv} = \{([a], [x]) \mid (a, x) \in G\}$. Let $\mathbb{S} = (\mathfrak{L}, \mathfrak{L}, S)$ be a basic Boolean joining-system such that $G^{\equiv} \subseteq S$. By Proposition 2 we know such basic Boolean joining-system always exist. Moreover, there must be a smallest basic Boolean joining space G_2 such that $G^{\equiv} \subseteq G_2$ and for every basic Boolean joining space S that extends G^{\equiv} , $G_2 \subseteq S$. Here we use the notation G_2 for the resemblance of basic input/output logic. Such G_2 is the basic Boolean joining space generated by G^{\equiv} . The following proposition shows how we can construct G_2 .

Proposition 3. *Let G'_2 be constructed as follows.⁵*

- $G_2^0 = G \equiv$
 - G_2^{i+1} contains all $([a], [x])$ for which:
 1. for some $([b], [y]) \in G_2^i$, $([b], [y]) \preceq ([a], [x])$
 2. there exist $([a], [y]), ([a], [z]) \in G_2^i$ such that $[x] = [y] \cdot [z]$
 3. there exist $([b], [x]), ([c], [x]) \in G_2^i$ such that $[a] = [b] + [c]$
- Let $G'_2 = \bigcup_{i=0}^{\infty} G_2^i$, then $G'_2 = G_2$.

With proposition 3 in hand, we now prove a correspondence result. Intuitively, this result states that every norm (a, x) is logically derivable from a set of norms G if and only if it is in the space of norms algebraically generated by G .

Theorem 2. *The following three propositions are equivalent:*

1. $(a, x) \in \text{deriv}_2(G)$.
2. $([a], [x]) \in G_2$.
3. $x \in \text{out}_2(G, a)$.

3.2 Input/output logics and joining-system

The previous subsection proves a correspondence between basic input/output logic and basic Boolean joining-system. We now prove a correspondence between other input/output logics and other Boolean joining-systems.

Definition 6 (Boolean joining-system). *Given a structure $\mathbb{S} = (\mathfrak{A}, \mathfrak{B}, S)$ where $\mathfrak{A}, \mathfrak{B}$ are Boolean algebras. Given the following conditions:*

1. If $(a, x) \in S$ and $(a, x) \preceq (b, y)$, then $(b, y) \in S$.
 2. If $(a, x) \in S$ and $(a, y) \in S$, then $(a, x \cdot_B y) \in S$.
 3. If $(a, x) \in S$ and $(b, x) \in S$, then $(a +_A b, x) \in S$.
 4. if $(a, x) \in S$ and $(a \cdot_A x, y) \in S$, then $(a, y) \in S$.
- If \mathbb{S} satisfies (1) and (2), then \mathbb{S} is a simple-minded Boolean joining-system. S is a simple-minded Boolean joining space of \mathbb{S} .
 - If \mathbb{S} satisfies (1), (2) and (3), then \mathbb{S} is a basic Boolean joining-system. S is a basic Boolean joining space of \mathbb{S} .
 - If \mathbb{S} satisfies (1), (2) and (4), then \mathbb{S} is a simple-minded reusable Boolean joining-system. S is a simple-minded reusable Boolean joining space of \mathbb{S} .
 - If \mathbb{S} satisfies (1), (2), (3) and (4), then \mathbb{S} is a basic reusable Boolean joining-system. S is a basic reusable Boolean joining space of \mathbb{S} .

Similar to Proposition 2, we prove the existence of the largest and the smallest simple-minded/basic/simple-minded reusable/basic reusable joining space.

⁵ The author thanks a reviewer of Journal of Logic and Computation for his/her contribution to this proposition.

Proposition 4. Given two Boolean algebras $\mathfrak{A}, \mathfrak{B}$,

1. $(\mathfrak{A}, \mathfrak{B}, A \times B)$ is a simple-minded/basic/simple-minded reusable/basic reusable Boolean joining-system.
2. If for all $i \in I$, $(\mathfrak{A}, \mathfrak{B}, S_i)$ is a simple-minded Boolean joining-system, then $(\mathfrak{A}, \mathfrak{B}, \bigcap_{i \in I} S_i)$ is a simple-minded Boolean joining-system. And similarly for basic/simple-minded reusable/basic reusable joining space.

Let G be a set of ordered pairs of formulas of \mathbb{L}_p . Let G_1 to G_4 be respectively the smallest simple-minded/ basic/simple-minded reusable/basic reusable joining space that extends G^{\equiv} . We have the the following correspondence result:

Theorem 3. For $i \in \{1, 2, 3, 4\}$, The following three propositions are equivalent:

1. $(a, x) \in deriv_i(G)$.
2. $([a], [x]) \in G_i$.
3. $x \in out_i(G, a)$.

4 Intuitionistic input/output logic and joining systems

A frequent belief about input/output logic is that it presupposes classical propositional logic. Parent *et al* [18] show that this is a misunderstanding by building input/output logic on top of intuitionistic logic. In this section, we show that there is an algebraic companion for intuitionistic input/output logic, the Heyting joining-system. To do this we first introduce intuitionistic input/output logic and Heyting joining-system, then construct the correspondence.

4.1 Intuitionistic input/output logic

Intuitionistic logic [27] is different from classical logic by omitting the principle of excluded middle and the *reductio ad absurdum* rule. Intuitionistic input/output logic is based on intuitionistic propositional logic (IPL), the propositional fragment of intuitionistic logic.

Given a set of propositional letters P , the language of intuitionistic propositional logic \mathbb{L}_I is defined as follows:

$$a, b ::= \perp \mid p \mid a \wedge b \mid a \vee b \mid a \rightarrow b$$

Here $p \in P$ and we use $\neg a$ as an abbreviation of $a \rightarrow \perp$.

A proof system of intuitionistic propositional logic is defined via the following Gentzen sequent calculus [18]:

– Group 1: Let A, B be finite set of formulas

- (Ref) If $a \in A$, then $A \vdash_I a$
- (Mon) $\frac{A \vdash_I a}{A \cup B \vdash_I a}$
- (Cut) $\frac{A \vdash_I a \quad A \cup \{a\} \vdash_I b}{A \vdash_I b}$

The labels (Ref) and (Mon) are mnemonic for “reflexivity” and “monotony” respectively.

- Group 2:
 - $\frac{A \vdash_I a \quad A \vdash_I b}{A \vdash_I a \wedge b} (\wedge:\text{I})$
 - $\frac{A \vdash_I a \wedge b}{A \vdash_I a} (\wedge:\text{E})$
 - $\frac{A \vdash_I a}{A \vdash_I a \vee b} (\vee:\text{I})$
 - $\frac{A \cup \{a\} \vdash_I c \quad A \cup \{b\} \vdash_I c}{A \vdash_I c} \quad A \vdash_I a \vee b (\vee:\text{E})$
 - $\frac{A \cup \{a\} \vdash_I b}{A \vdash_I a \rightarrow b} (\rightarrow:\text{I})$
 - $\frac{A \vdash_I a \quad A \vdash_I a \rightarrow b}{A \vdash_I b} (\rightarrow:\text{E})$
 - $\frac{A \vdash_I \perp}{A \vdash_I a} (\perp:\text{E})$

If $\emptyset \vdash_I a$ then we say a is provable in IPL. If A is infinite, then we let $A \vdash_I a$ iff $A' \vdash_I a$ for some finite $A' \subseteq A$.

Let G be a set of ordered pairs of formulas of \mathbb{L}_I . For a set of formulas $A \subseteq \mathbb{L}_I$, let $Cn^I(A) = \{a \in \mathbb{L}_I \mid A \vdash_I a\}$. To define intuitionistic input/output logic the concept of saturated set is needed.

Definition 7 (Saturated set [25]). *A set $A \subseteq \mathbb{L}_I$ is said to be saturated if the following three conditions hold:*

1. $A \not\vdash_I \perp$
2. if $a \vee b \in A$ then $a \in A$ or $b \in A$
3. if $A \vdash_I a$ then $a \in A$

Parent *et al* [18] define input/output logic based on intuitionistic logic as follows:

- $out_1^I(G, A) = Cn^I(G(Cn^I(A)))$
- $out_2^I(G, A) = \bigcap \{Cn^I(G(B)) \mid A \subseteq B, B \text{ is saturated or } B = \mathbb{L}_I\}$
- $out_3^I(G, A) = \bigcap \{Cn^I(G(B)) \mid A \cup G(B) \subseteq B = Cn^I(B)\}$
- $out_4^I(G, A) = \bigcap \{Cn^I(G(B)) \mid A \cup G(B) \subseteq B, B \text{ is saturated or } B = \mathbb{L}_I\}$

The proof system of intuitionistic input/output logics are similar to its propositional counterpart. Parent *et al* [18] use AND, OR, CT and the intuitionistic version of SI and WO:

- SI^I (intuitionistic strengthening the input): from (a, x) to (b, x) whenever $b \vdash_I a$
- WO^I (intuitionistic weakening the output): from (a, x) to (a, y) whenever $x \vdash_I y$

The derivation system based on the rules SI_I , WO_I and AND is called $deriv_1^I$. Adding OR and CT to $deriv_1^I$ gives $deriv_2^I$ and $deriv_3^I$ respectively. The five rules together define $deriv_4^I$. In Parent *et al* [18], the following completeness theorems are given:

Theorem 4 ([18]). *Given an arbitrary normative system G and a set A of formulas,*

- For $i \in \{1, 2, 3\}$, $x \in out_i^I(G, A)$ iff $x \in deriv_i^I(G, A)$.
- If $x \in deriv_4^I(G, A)$, then $x \in out_4^I(G, A)$.⁶

⁶ It is an open problem whether the other direction of the implication holds.

4.2 Heyting joining-system

Definition 8. A Heyting algebra is a partially ordered set $(H, 0, 1, \leq, \cdot, +, \rightarrow)$ with a smallest element 0 , a largest element 1 and three operations \cdot , $+$ and \rightarrow satisfying the following conditions, for all $x, y, z \in H$

1. $x \leq 1$
2. $x \cdot y \leq x$
3. $x \cdot y \leq y$
4. $z \leq x$ and $z \leq y$ implies $z \leq x \cdot y$
5. $0 \leq x$
6. $x \leq x + y$
7. $y \leq x + y$
8. $x \leq z$ and $y \leq z$ implies $x + y \leq z$
9. $z \leq (x \rightarrow y)$ iff $z \cdot x \leq y$

A valuation from IPL to a Heyting algebra is a function $V : P \rightarrow H$. We extend V to arbitrary formulas by putting:

$$\begin{aligned} V(\perp) &= 0 \\ V(\top) &= 1 \\ V(a \wedge b) &= V(a) \cdot V(b) \\ V(a \vee b) &= V(a) + V(b) \\ V(a \rightarrow b) &= V(a) \rightarrow V(b) \end{aligned}$$

A formula a is said to be H-valid if $V(a) = 1$ for all valuations V .

Theorem 5 ([26]). A formula a is provable in IPL iff a is H-valid.

Heyting joining-systems are defined in a similar way to Boolean joining-systems.

Definition 9 (Heyting joining-system). A Heyting joining-system is a structure $\mathbb{S} = (\mathfrak{A}, \mathfrak{B}, S)$ such that $\mathfrak{A} = (A, 0_A, 1_A, \leq_A, \cdot_A, +_A, \rightarrow_A)$, $\mathfrak{B} = (B, 0_B, 1_B, \leq_B, \cdot_B, +_B, \rightarrow_B)$ are Heyting algebras and $S \subseteq A \times B$ satisfies certain conditions:

1. If $(a, x) \in S$ and $(a, x) \preceq (b, y)$, then $(b, y) \in S$.
2. If $(a, x) \in S$ and $(a, y) \in S$, then $(a, x \cdot_B y) \in S$.
3. If $(a, x) \in S$ and $(b, x) \in S$, then $(a +_A b, x) \in S$.
4. If $(a, x) \in S$ and $(a \cdot_A x, y) \in S$, then $(a, y) \in S$.

The Heyting joining-system satisfies 1 and 2 is called simple-minded. Adding 3 and 4 produces basic and simple-minded reusable Heyting joining-system respectively. The four condition together give rise to basic reusable Heyting joining-system.

4.3 Correspondence

Let \equiv_I be the provable equivalence relation on \mathbb{L}_I , i.e. for every formula $\phi, \psi \in \mathbb{L}_I$, $\phi \equiv_I \psi$ iff $\vdash_I \phi \leftrightarrow \psi$. Let $\mathbb{L}_I^{\equiv_I}$ be the equivalence classes that \equiv_I induces on \mathbb{L}_I . For any formula $\phi \in \mathbb{L}_I$, let $[\phi]^I$ denote the equivalence class containing ϕ .

Definition 10 (Intuitionistic Lindenbaum-Tarski algebra). *The intuitionistic Lindenbaum-Tarski algebra for \mathbb{L}_I is a structure $(\mathbb{L}_I^{\equiv_I}, 0, 1, \leq, +, \cdot, \rightarrow)$ where $0 = [\perp]^I$, $1 = [\top]^I$, $[\phi]^I \leq [\psi]^I$ iff $[\phi]^I \cdot [\psi]^I = [\phi]^I$, $[\phi]^I + [\psi]^I = [\phi \vee \psi]^I$, $[\phi]^I \cdot [\psi]^I = [\phi \wedge \psi]^I$, $[\phi]^I \rightarrow [\psi]^I = [\phi \rightarrow \psi]^I$.*

It can be verified that an intuitionistic Lindenbaum-Tarski algebra is a Heyting algebra. Let G be a set of ordered pairs of formulas of \mathbb{L}_I . Let $G^{\equiv_I} = \{([a]^I, [x]^I) \mid (a, x) \in G\}$, and G_1^I to G_4^I be the simple-minded/basic/simple-minded reusable/basic reusable Heyting joining-system generated by G^{\equiv_I} respectively. Then we have the following correspondence result between intuitionistic input/output logics and Heyting joining-system:

Theorem 6. *For $i \in \{1, 2, 3\}$, the following three statements are equivalent:*

1. $(a, x) \in \text{deriv}_i^I(G)$
2. $([a], [x]) \in G_i^I$
3. $x \in \text{out}_i^I(G, a)$

5 Quasi input/output logic and Boolean quasi-ordering

The algebraic structure Lindahl and Odelstad [7] use for their theory of joining-system are Boolean quasi-ordering. The logical companion of Boolean quasi-ordering has not been developed yet. In this section, we fill this gap by building quasi input/output logic.

5.1 Boolean quasi-ordering

Definition 11 (Boolean quasi-ordering [9]). *A Boolean quasi-ordering is a structure $(B, \perp, \top, \wedge, \vee, ', R)$ such that $(B, \perp, \top, \wedge, \vee, ')$ is a Boolean algebra, and $R \subseteq B \times B$ is a binary, reflexive and transitive relation on B which satisfies the following conditions for all a, b and c in B :*

- aRb and aRc implies $aR(b \wedge c)$
- aRb implies $b'Ra'$
- $(a \wedge b)Ra$
- not $\top R \perp$

A reflexive and transitive relation is called a quasi-ordering in Lindahl and Odelstad [7]. Various joining-system based on Boolean quasi-ordering are defined as follows:

Definition 12 (Quasi joining-system). A quasi joining-system is a structure $\mathbb{S} = (\mathfrak{A}, \mathfrak{B}, S)$ such that $\mathfrak{A} = (A, \perp_A, \top_A, \wedge_A, \vee_A, \prime_A, R_A)$, $\mathfrak{B} = (B, \perp_B, \top_B, \wedge_B, \vee_B, \prime_B, R_B)$ are Boolean quasi-orderings and $S \subseteq A \times B$ satisfies certain conditions:

1. If $(a, x) \in S$ and bR_Aa and xR_By , then $(b, y) \in S$.
2. If $(a, x) \in S$ and $(a, y) \in S$, then $(a, x \wedge_B y) \in S$
3. If $(a, x) \in S$ and $(b, x) \in S$, then $(a \vee_B a, x) \in S$
4. if $(a, x) \in S$ and $(a \wedge_A x, y) \in S$, then $(a, y) \in S$.

The quasi joining-system satisfies 1 and 2 is called simple-minded. Adding 3 and 4 produces basic and simple-minded reusable quasi joining-system respectively. The four conditions together give basic reusable quasi joining-system.

5.2 Quasi input/output logic

Given a set P of propositional letters, \mathbb{L}_Q is the propositional language based on P . Let $p \in P$,

$$a, b := \top \mid p \mid \neg a \mid a \wedge b \mid a \vee b$$

Analog to Boolean quasi-ordering, we define a quasi entailment $\vdash_q \subseteq \mathbb{L}_Q \times \mathbb{L}_Q$ to be a reflexive and transitive relation satisfying the following:

- $a \vdash_q b$ and $a \vdash_q c$ implies $a \vdash_q b \wedge c$
- $a \vdash_q b$ implies $\neg b \vdash_q \neg a$
- $a \wedge b \vdash_q a$
- $a \wedge b \vdash_q b$
- $\top \not\vdash_q \perp$

We further lift \vdash_q to $\vdash_Q \subseteq 2^{\mathbb{L}_Q} \times \mathbb{L}_Q$ such that $A \vdash_Q b$ iff there exist $a_1, \dots, a_n \in A$ such that $a_1 \wedge \dots \wedge a_n \vdash_q b$. For a set of formulas $A \subseteq \mathbb{L}_Q$, we let $Cn^Q(A) = \{a \in \mathbb{L}_Q \mid A \vdash_Q a\}$.

The following proposition shows that \vdash_Q is reflexive, transitive, monotonic and closed under conjunction.

Proposition 5. Every quasi entailment \vdash_Q has the following properties:

1. $A \vdash_Q a$ for every $a \in A$.
2. If $A \vdash_Q a$ and $\{a\} \vdash_Q b$, then $A \vdash_Q b$.
3. If $A \vdash_Q a$ then $A \cup B \vdash_Q a$.
4. If $A \vdash_Q a$ and $A \vdash_Q b$, then $A \vdash_Q a \wedge b$.

Given a quasi entailment \vdash_Q , we define quasi input/output logic out_1^Q to out_4^Q as following:

- $out_1^Q(G, A) = Cn^Q(G(Cn^Q(A)))$
- $out_2^Q(G, A) = \bigcap \{Cn^Q(G(B)) : A \subseteq B = Cn^Q(B), B \text{ is disjunctive}\}$
- $out_3^Q(G, A) = \bigcap \{Cn^Q(G(B)) : A \cup G(B) \subseteq B = Cn^Q(B)\}$
- $out_4^Q(G, A) = \bigcap \{Cn^Q(G(B)) : A \cup G(B) \subseteq B = Cn^Q(B), B \text{ is disjunctive}\}$

Here a set of formulas B is disjunctive if $a \vee b \in B$ implies either $a \in B$ or $b \in B$. $out_1^Q(G, A)$ to $out_4^Q(G, A)$ are called *simple-minded quasi output*, *basic quasi output*, *simple-minded reusable quasi output* and *basic reusable quasi output* respectively.

The proof system of quasi input/output logics is similar to its propositional and intuitionistic counterpart. We use AND, OR, CT and the quasi version of SI and WO:

- SI^Q (quasi strengthening the input): from (a, x) to (b, x) whenever $b \vdash_Q a$
- WO^Q (quasi weakening the output): from (a, x) to (a, y) whenever $x \vdash_Q y$

The derivation system based on the rules SI^Q , WO^Q and AND is called $deriv_1^Q$. Adding OR and CT to $deriv_1^Q$ gives $deriv_2^Q$ and $deriv_3^Q$ respectively. The five rules together defines $deriv_4^Q$.

Now we turn to the completeness theorem. We prove the case for out_1^Q and out_3^Q . out_2^Q and out_4^Q are harder to handle. Especially for out_4^Q , the fixed point theoretic technique used for out_3^Q cannot be applied to out_4^Q in a straightforward way. We therefore leave them for future work.

Theorem 7. *Given an arbitrary normative system G ,*

$$(a, x) \in deriv_1^Q(G) \text{ iff } x \in out_1^Q(G, a)$$

For the case of out_3^Q , we use a fixed point theoretic technique. Such technique is versatile to build reusable input/output logics [22]. We define a function $f_A^G : 2^{\mathbb{L}_Q} \rightarrow 2^{\mathbb{L}_Q}$ such that $f_A^G(X) = Cn^Q(A \cup G(X))$. It can be proved that f_A^G is monotonic with respect to the set theoretical \subseteq relation, and $(2^{\mathbb{L}_Q}, \subseteq)$ is a complete lattice. Then by Tarski's fixed point theorem [24] there exist a least fixed point of f_A^G . Here a least fixed point of f_A^G is a set $X \subseteq \mathbb{L}_Q$ such that $f_A^G(X) = X$ and for all $Y \subseteq \mathbb{L}_Q$, if $f_A^G(Y) = Y$ then $X \subseteq Y$. Let B_A^G be the least fixed point of the function f_A^G . The following lemma shows how the least fixed point can be constructed.

Lemma 1. $B_A^G = \bigcup_{i=0}^{\infty} B_{A,i}^G$, where $B_{A,0}^G = Cn^Q(A)$, $B_{A,i+1}^G = Cn^Q(A \cup G(B_{A,i}^G))$.

When the set $A = \{a\}$ is a singleton, we use B_a^G as an abbreviation of $B_{\{a\}}^G$. The semantics of out_3^Q can equivalently be defined by the least fixed point of f_A^G .

Theorem 8. *Given an arbitrary normative system G ,*

$$x \in Cn^Q(G(B_a^G)) \text{ iff } x \in out_3^Q(G, a)$$

Theorem 9 (Soundness). *If $(a, x) \in deriv_3^Q(G)$ then $x \in Cn^Q(G(B_a^G))$*

Theorem 10 (Completeness). *If $x \in Cn^Q(G(B_a^G))$, then $(a, x) \in deriv_3^Q(G)$.*

5.3 Correspondence

Let \equiv_Q be the provable equivalence relation on \vdash_Q , i.e. for every formula $\phi, \psi \in \mathbb{L}_Q$, $\phi \equiv_Q \psi$ iff $\vdash_Q \phi \leftrightarrow \psi$. Let $\mathbb{L}_Q^{\equiv_Q}$ be the equivalence classes that \equiv_Q induces on \mathbb{L}_Q . For any formula $\phi \in \mathbb{L}_Q$, let $[\phi]^Q$ denote the equivalence class containing ϕ .

Definition 13 (Quasi Lindenbaum-Tarski algebra). *The quasi Lindenbaum-Tarski algebra for a quasi logic \mathbb{L}_Q is a structure $(\mathbb{L}_Q^{\equiv_Q}, 0, 1, +, \cdot, -, \leq)$ where $0 = [\perp]^Q$, $1 = [\top]^Q$, $[a]^Q + [b]^Q = [a \vee b]^Q$, $[a]^Q \cdot [b]^Q = [a \wedge b]^Q$, $-[a] = [-a]^Q$, $[a]^Q \leq [b]^Q$ iff $a \vdash_Q b$.*

It can be verified that quasi Lindenbaum-Tarski algebra is a Boolean quasi-ordering. Let $G^{\equiv_Q} = \{([a]^Q, [x]^Q) | (a, x) \in G\}$, G_1^Q and G_3^Q be the simple-minded/simple-minded reusable quasi joining-system generated by G^{\equiv_Q} respectively. Then we have the following correspondence between quasi input/output logics and quasi joining-system:

Theorem 11. *The following three propositions are equivalent: for $i \in \{1, 3\}$*

- 1 $(a, x) \in \text{deriv}_i^Q(G)$
- 2 $([a], [x]) \in G_i^Q$
- 3 $x \in \text{out}_i^Q(G, a)$

6 Holistic perspectives on normative systems

In this section, we discuss some insights obtained from the algebraic approach to normative systems. As stated in the Section 1. The algebraic framework gives us a holistic view on normative systems.

One feature of the input/output framework is that it adopts operational rather than possible world semantics. There is no exterior structure in such operational semantics. Therefore tools to compare the similarity of structures, like bi-simulation and isomorphism, play no role in input/output logic. This feature makes it difficult to analyze the *similarity* of normative systems using input/output logic, although the *equivalence* of normative systems can be represented within the input/output framework [5]. Such limitation is an opportunity for the algebraic framework because it is able to discuss the similarity of normative systems by extending isomorphism of an algebra to isomorphism of the joining-system based on the algebra.

6.1 Similarity of normative systems

For two algebraic structures A and B , if they are isomorphic then they are essentially the same. We can extend the isomorphism of Boolean algebra to Boolean joining-system.

Definition 14 (Isomorphism of Boolean algebra). For two Boolean algebras $\mathfrak{A} = (A, +, \cdot, -, 0, 1)$ and $\mathfrak{A}' = (A', +, \cdot, -, 0, 1)$ and h a map from A to A' . We say that h is an isomorphism from \mathfrak{A} to \mathfrak{A}' iff for any $x, y \in A$, h satisfies the following conditions:

1. h is bijective
2. $h(x + y) = h(x) + h(y)$
3. $h(x \cdot y) = h(x) \cdot h(y)$
4. $h(1) = 1$

Given an isomorphism h from \mathfrak{A} to \mathfrak{A}' , it is easy to check that for all $x, y \in A$ and $x', y' \in A'$, if $h(x) = x'$ and $h(y) = y'$, then $x \leq y$ iff $x' \leq y'$. Now we extend isomorphism to Boolean joining-system.

Definition 15 (Isomorphism of joining-system). For two Boolean joining-system $\mathfrak{S} = (\mathfrak{A}, \mathfrak{B}, S)$ and $\mathfrak{S}' = (\mathfrak{A}', \mathfrak{B}', S')$ and h a map from $A \cup B$ to $A' \cup B'$. We say that h is an isomorphism from \mathfrak{S} to \mathfrak{S}' iff h satisfies the following conditions:

1. h is bijective
2. the restriction of h on A is an isomorphism from A to A'
3. the restriction of h on B is an isomorphism from B to B'
4. $(a, x) \in S$ iff $(h(a), h(x)) \in S'$

If there exist an isomorphism from \mathfrak{S} to \mathfrak{S}' , then we say \mathfrak{S} and \mathfrak{S}' are isomorphic. Two isomorphic joining-system can naturally be understood as structurally the same.

In political philosophy, research on totalitarianism [2, 3] views the ideology of Nazi Germany and Soviet Union to be similar. Stalinism and Nazism are described as “totalitarian twins”. Using the algebraic representation of normative systems we can describe such similarity in a mathematical flavor.

Example 1. Let H be “you worship Hitler”, S be “you respect Stalin”, N be “you are a member of the Nazi Party”, C be “you are a member of the communist party”, R be “you are against to the rich people”, J be “you hate Jews”. Let B_1 be the Boolean algebra generated by $\{H, N, J\}$, B_2 be the Boolean algebra generated by $\{S, C, R\}$. Let $N_1 = \{(\top, H), (N, J)\}$ be the normative system saying “you are obligatory to worship Hitler” and “you are obligatory to hate Jews, given the condition that you are a member of the Nazi Party”. Let $N_2 = \{(\top, S), (C, R)\}$ be the normative system saying “you are obligatory to respect Stalin” and “you are obligatory to be against to the rich people, given the condition that you are a member of the communist party”. For the joining-system (B_1, B_1, N_1) generated by N_1 and (B_2, B_2, N_2) generated by N_2 , we can build an isomorphism h such that $h(H) = S, h(N) = C, h(J) = R$. We therefore conclude that N_1 and N_2 are similar.

Not only isomorphism can be used as an algebraic tool to analyze the similarity of normative systems, embedding is also a useful tool.

Definition 16 (Embedding of joining-system). For two joining-system $\mathbb{S} = (A, B, S)$ and $\mathbb{S}' = (A', B', S')$ and h a map from $A \cup B$ to $A' \cup B'$. We say that h is an embedding from \mathbb{S} to \mathbb{S}' iff h satisfies the following conditions:

1. h is injective
2. the restriction of h on A is an isomorphism from A to $h(A)$
3. the restriction of h on B is an isomorphism from B to $h(B)$
4. if $(a, x) \in S$ then $(h(a), h(x)) \in S'$

In political philosophy, totalitarianism is generally viewed an extreme version of authoritarianism [21]. That means a normative system of totalitarianism can be considered as an extension of a normative system of authoritarianism. Therefore mathematically there should be an embedding from the normative system of authoritarianism to the normative system of totalitarianism. The following is an illustration.

Example 2. Let B_1, N_1 be same as in the previous example. Let G be “you like Gaddafi”, F be “you are a follower of Gaddafi”, A be “you dislike America”. Let B_3 be the Boolean algebra generated by $\{G, F, A\}$. Let $N_3 = \{\top, G\}$ be the normative system saying “you are obligatory to like Gaddafi”. For the joining-system (B_1, B_1, N_1) generated by N_1 and (B_3, B_3, N_3) generated by N_3 , we can build an embedding h such that $h(G) = H, h(F) = N, h(A) = J$. We therefore say that N_1 is an extension of N_3 .

It must be confessed that our examples greatly simplify the complexity of political philosophy to a degree that lots of valuable information is lost. However, we still believe that the algebraic approach offers a useful mathematical tool for political philosophy and ethics in the sense that more complex normative systems can also be characterized by joining-system. For example, we may let a normative system contains three subsystems: constitutive norms, obligatory norms and permissive norms. Such a complex normative system can still be represented by input/output logic [23], therefore we can develop more complicated joining-system to algebraically characterize complex normative systems. Using more complicated joining-system we can discuss the similarity of different ideology without losing too much valuable information.

6.2 The core of a normative system

In section 2.2 the narrowness relation \preceq is defined as $(a, x) \preceq (b, y)$ iff $b \leq_A a$ and $x \leq_B y$. We now further define the strict narrowness relation \prec as $(a, x) \prec (b, y)$ iff $(a, x) \preceq (b, y)$ and not $(b, y) \preceq (a, x)$. Intuitively, if $(a, x) \prec (b, y)$ then (a, x) is logically stronger than (b, y) in the sense that if (a, x) is in a joining-system, then (b, y) must also be in the joining-system, but not vice versa.

We now use such strict narrowness relation to define the core of a normative system. A norm (a, x) is minimal in a joining-system $\mathbb{S} = (\mathfrak{A}, \mathfrak{B}, S)$ iff there is no $(b, y) \in S$ such that $(b, y) \prec (a, x)$. In Odelstad and Lindahl [17], such a minimal norm is called a connection from \mathfrak{A} to \mathfrak{B} . As noticed by Odelstad and Lindahl

[17], the set of all minimal elements of a joining-system can be viewed as the core of the system in the sense that the whole system is uniquely determined by its minimal norms. Therefore we can logically deduce the whole system, if we know the core of the system. For a joining-system \mathbb{S} , let $core(\mathbb{S}) = \{(a, x) \in S \mid (a, x) \text{ is minimal in } \mathbb{S}\}$ denote the set of all its minimal norms. The following are formal statements about the properties of the core of finite joining-system.

Observation 1 *For every joining-system $\mathbb{S} = (\mathfrak{A}, \mathfrak{B}, S)$, if S is finite then $core(\mathbb{S}) \neq \emptyset$.*

Observation 2 *For all joining-system $\mathbb{S} = (\mathfrak{A}, \mathfrak{B}, S)$, if S is finite, then for any $(a, x) \in S$, there exists $(b, y) \in core(\mathbb{S})$ such that $(b, y) \preceq (a, x)$.*

Observation 3 *For any joining-system $\mathbb{S} = (\mathfrak{A}, \mathfrak{B}, S)$ and $\mathbb{S}' = (\mathfrak{A}, \mathfrak{B}, S')$, if both S and S' are finite, then $core(\mathbb{S}) = core(\mathbb{S}')$ iff $S = S'$.*

All the three observations work only for finite joining-system. Observation 1 states that the core always exist. Observation 3 states that the whole system is determined by the core. Regarding observation 2, it states that for any norms which is not in the core, there is a norm in the core which is stronger and therefore able to generate it. Observation 2 partly answers the problem of norms redundancy, which is raised by Boella *et al* [6] and addressed by van der Torre [28]. According to observation 2, all norms which are not in the core are redundant.

7 Conclusion and future work

The main contribution of this paper is the correspondence results between input/output logic and joining-system. We present three versions of correspondence: input/output logic and Boolean joining-system, intuitionistic input/output logic and Heyting joining-system, quasi input/output logic and quasi joining-system. Quasi input/output logic does not exist before this paper. We develop it and prove the completeness theorem of simple-minded and simple-minded reusable quasi input/output logic. We use a fixed point theoretical technique to prove the completeness of simple-minded reusable quasi input/output logic.

The correspondence results illustrate that normative systems can be analyzed using three different tools: proof theory, semantics and algebra. Each tool gives us some insight into normative systems. The algebraic tool offers a holistic perspective on normative systems. We use isomorphism and embedding of joining-system to discuss the similarity of normative systems.

There are a lot of future work to be done. A natural direction is to build basic and basic reusable quasi input/output logic, and construct the correspondence result. Another direction is to use more advanced logic and algebra, say Boolean algebra with modal operators, to relate input/output logic and joining-system.

Acknowledgments

The author thanks Leendert van der Torre, Xavier Parent, three anonymous reviewers of the LORI committee, two reviewers of the Journal of Logic and Computation for their valuable suggestions and comments.

References

1. Carlos E. Alchourron and Eugenio Bulygin. *Normative Systems*. Springer-Verlag, Wien New York, 1971.
2. Hannah Arendt. *The Origins of Totalitarianism*. The World Publishing Company, Cleveland and New York, 1958.
3. John A. Armstrong. *The Politics of Totalitarianism*. Random House, New York, 1961.
4. Patric Blackburn, Maarten De Rijke, and Yde Venema. *Modal logic*. Cambridge University Press, Cambridge, 2001.
5. Guido Boella, Jan Broersen, and Leendert van der Torre. Reasoning about constitutive norms, counts-as conditionals, institutions, deadlines and violations. In The Duy Bui, Tuong Vinh Ho, and Quang-Thuy Ha, editors, *PRIMA*, volume 5357 of *Lecture Notes in Computer Science*, pages 86–97. Springer, 2008.
6. Guido Boella, Leendert van der Torre, and Harko Verhagen. Introduction to the special issue on normative multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 17(1):1–10, 2008.
7. Lars Lindahl and Jan Odelstad. An algebraic analysis of normative systems. *Ratio Juris*, 13:261–278, 2000.
8. Lars Lindahl and Jan Odelstad. Intermediaries and intervenients in normative systems. *Journal of applied logic*, pages 229–250, 2008.
9. Lars Lindahl and Jan Odelstad. TJS. a formal framework for normative systems with intermediaries. In John Horty, Dov Gabbay, Xavier Parent, Ron van der Meyden, and Leendert van der Torre, editors, *Handbook of Deontic Logic and Normative Systems*. College Publications, 2013.
10. David Makinson. On a fundamental problem of deontic logic. In P. McNamara and H. Prakken, editors, *Norms, Logics and Information Systems*, pages 29–53. IOS Press, Amsterdam, 1999.
11. David Makinson and Leendert van der Torre. Input-output logics. *Journal of Philosophical Logic*, 29:383–408, 2000.
12. David Makinson and Leendert van der Torre. Constraints for input/output logics. *Journal of Philosophical Logic*, 30(2):155–185, 2001.
13. David Makinson and Leendert van der Torre. Permission from an input/output perspective. *Journal of Philosophical Logic*, 32:391–416, 2003.
14. David Makinson and Leendert van der Torre. What is input/output logic? In B. Lowe, W. Malzkorn, and T. Rasch, editors, *Foundations of the Formal Sciences II: Applications of Mathematical Logic in Philosophy and Linguistics*, pages 163–174, 2003.
15. Jan Odelstad and Magnus Boman. Algebras for agent norm-regulation. *Annals of Mathematics and Artificial Intelligence*, 42:141–166, 2004.
16. Jan Odelstad and Lars Lindahl. Normative systems represented by Boolean quasi-orderings. *Nordic Journal of Philosophical logic*, 5:161–174, 2000.
17. Jan Odelstad and Lars Lindahl. The role of connections as minimal norms in normative systems. In T. Bench-Capon, A. Daskalopulu, and R. Winkels, editors, *Legal Knowledge and Information Systems*. IOS Press, Amsterdam, 2002.

18. Xavier Parent, Dov Gabbay, and Leendert van der Torre. An intuitionistic basis for input/output logic. In Sven Ove Hasson, editor, *David Makinson on Classical Methods for Non-Classical Problems*. Springer, 2012.
19. Xavier Parent and Leendert van der Torre. I/O logic. In John Horty, Dov Gabbay, Xavier Parent, Ron van der Meyden, and Leendert van der Torre, editors, *Handbook of Deontic Logic and Normative Systems*. College Publications, 2013.
20. Xavier Parent and Leendert van der Torre. “Sing and dance!” - input/output logics without weakening. In Fabrizio Cariani, Davide Grossi, Joke Meheus, and Xavier Parent, editors, *Deontic Logic and Normative Systems - 12th International Conference, DEON 2014, Ghent, Belgium, July 12-15, 2014. Proceedings*, volume 8554 of *Lecture Notes in Computer Science*, pages 149–165. Springer, 2014.
21. Paul C. Sondrol. Totalitarian and authoritarian dictators: A comparison of fidel castro and alfredo stroessner. *Journal of Latin American Studies*, 23(3), 2009.
22. Xin Sun. *How to build input/output logics*, 2014. Manuscript.
23. Xin Sun and Leendert van der Torre. Combining regulative and constitutive norms in input/output logic. In *Proceedings of 12th International Conference on Deontic logic and Normative Systems (DEON 2014)*, 2014.
24. Alfred Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5(2):285–309, 1955.
25. Richmond H. Thomason. On the strong semantical completeness of the intuitionistic predicate calculus. *Journal of Symbolic Logic*, 33(1):1–7, 1968.
26. Anne Sjerp Troelstra and Dirk van Dalen. *Constructivism in Mathematics: An Introduction*. North Holland, Amsterdam, 1988.
27. Dirk van Dalen. Intuitionistic logic. In *Handbook of philosophical logic*, pages 225–339. Springer, 1986.
28. Leendert van der Torre. Deontic redundancy: A fundamental challenge for deontic logic. In Guido Governatori and Giovanni Sartor, editors, *DEON*, volume 6181 of *Lecture Notes in Computer Science*, pages 11–32. Springer, 2010.

Appendix

Proposition 1 Given a structure $\mathbb{S} = (\mathfrak{A}, \mathfrak{B}, S)$, where $\mathfrak{A}, \mathfrak{B}$ are Boolean algebras and $S \subseteq A \times B$, S is a basic Boolean joining space of \mathbb{S} if and only if it satisfies the following conditions:

1. For every finite set $X \subseteq B$ and every $a \in A$, if for every $x \in X$, $(a, x) \in S$, then for every $y \in F_{\uparrow}(X)$, $(a, y) \in S$.
2. For every finite set $X \subseteq A$ and every $a \in A$, if for every $x \in X$, $(x, b) \in S$, then for every $y \in I_{\downarrow}(X)$, $(y, b) \in S$.

Proof. Assume S is the basic Boolean joining space of \mathbb{S} . For the first condition, let X be an arbitrary finite subset of B . Without loss of generality, we can let $X = \{x_1, \dots, x_n\}$. Suppose $\forall x \in X$, $(a, x) \in S$. Then by applying clause 2' of Definition 2 finitely many times we have $(a, x_1 \cdot_B \dots \cdot_B x_n) \in S$. Since for all $y \in F_{\uparrow}(X)$, $x_1 \cdot_B \dots \cdot_B x_n \leq_B y$, we then know $(a, x_1 \cdot_B \dots \cdot_B x_n) \preceq (a, y)$. Therefore $(a, y) \in S$ by Definition 2. Similarly we can prove that the second condition is satisfied.

Now assume S satisfies the two conditions in this proposition. Assume $(a, x) \in S$ and $(a, x) \preceq (b, y)$, then $x \leq_B y$ and $y \in F_{\uparrow}(x)$, hence $(a, y) \in S$. Moreover we have $b \leq_A a$ and $b \in I_{\downarrow}(a)$, so we have $(b, y) \in S$. Assume $(a, x) \in S$ and $(a, y) \in S$. Since $x \cdot_B y \in F_{\uparrow}(\{x, y\})$, we know $(a, x \cdot_B y) \in S$. Similarly we can prove if $(a, x) \in S$ and $(b, x) \in S$, then $(a +_A b, x) \in S$. Therefore S is a basic joining space. \square

Proposition 2 Given two Boolean algebras $\mathfrak{A}, \mathfrak{B}$,

1. $(\mathfrak{A}, \mathfrak{B}, A \times B)$ is a basic Boolean joining-system.
2. If for all $i \in I$, $(\mathfrak{A}, \mathfrak{B}, S_i)$ is a basic Boolean joining-system, then $(\mathfrak{A}, \mathfrak{B}, \cap_{i \in I} S_i)$ is a basic Boolean joining-system.

Proof. The first item is safely left to readers. Here we prove the second item. For every finite set $X \subseteq A$, if for every $x \in X$, $(x, b) \in \cap_{i \in I} S_i$, then $(x, b) \in S_i$ for every $i \in I$. Therefore by Proposition 1 we have $\forall y \in I_{\downarrow}(X)$, $(y, b) \in S_i$. So we must have $(y, b) \in \cap_{i \in I} S_i$. Similarly we can prove the first item of Proposition 1. Therefore $\cap_{i \in I} S_i$ is a joining space of $\mathfrak{A} \times \mathfrak{B}$. \square

Proposition 3 Let G'_2 be constructed as follows.

- $G'_2{}^0 = G^{\equiv}$
 - $G'_2{}^{i+1}$ contains all $([a], [x])$ for which:
 1. for some $([b], [y]) \in G'_2{}^i$, $([b], [y]) \preceq ([a], [x])$
 2. there exist $([a], [y]), ([a], [z]) \in G'_2{}^i$ such that $[x] = [y] \cdot [z]$
 3. there exist $([b], [x]), ([c], [x]) \in G'_2{}^i$ such that $[a] = [b] + [c]$
- Let $G'_2 = \bigcup_{i=0}^{\infty} G'_2{}^i$, then $G'_2 = G_2$.

Proof. We have to show three things: (a) $G^\equiv \subseteq G'_2$. (b) G'_2 is a basic Boolean joining space. (c) for all $G'' \subsetneq G'_2$, G'' is not a basic Boolean joining space or $G^\equiv \not\subseteq G''$.

- (a) This is obvious in view of the construction.
- (b) We show that 1, 2' and 3' from Definition 2 hold for G' .
- Let $([a], [x]) \in G'_2$ and $([a], [x]) \preceq ([b], [y])$. Hence there is $i \geq 0$ such that $([a], [x]) \in G_2^i$. By (1), $([b], [y]) \in G_2^{i+1} \subseteq G'_2$.
 - Let $([a], [x]), ([b], [y]) \in G'_2$. Hence there is $i, j \geq 0$ such that $([a], [x]) \in G_2^i$ and $([b], [y]) \in G_2^j$. Let $k = \max\{i, j\}$. Then $([a], [x]), ([b], [y]) \in G_2^k$. By (2), $([a], [x] \cdot [y]) \in G_2^{k+1} \subseteq G'_2$.
 - 3' is shown analogously.
- (c) Let $G'' \subsetneq G'_2$. Suppose $G^\equiv \subseteq G''$. Hence, there is a minimal $i \geq 1$ such that $G_2^i \cap (G'_2 - G'') \neq \emptyset$. Let $([a], [x]) \in G_2^i \cap (G'_2 - G'')$. We can distinguish the cases (1), (2) and (3).
- According to (1), there is a $([b], [y]) \in G_2^{i-1}$ for which $([b], [y]) \preceq ([a], [x])$. Hence, by the minimality of i , $([b], [y]) \in G''$. But then G'' is not a joining space by Definition 2.
- Similarly we can show that in case of (2) and (3) G'' is not a joining space. This shows that G'_2 is a minimal joining space that extends G^\equiv and hence $G'_2 = G_2$.

□

Theorem 2 The following three propositions are equivalent:

1. $(a, x) \in \text{deriv}_2(G)$.
2. $([a], [x]) \in G_2$.
3. $x \in \text{out}_2(G, a)$.

Proof. $1 \Rightarrow 2$: This can be proved simply by induction on the length of derivation.

$2 \Rightarrow 3$: Assume $([a], [x]) \in G_2$. Hence there is an $i \geq 0$ such that $([a], [x]) \in G_2^i$ (see Proposition 3). We show that for each $([a], [x]) \in G_2$, $x \in \text{out}_2(G, a)$ by induction over i . For the induction base let $([a], [x]) \in G_2^0 = G^\equiv$. Trivially $x \in \text{out}_2(G, a)$. For the inductive cases, assume the conclusion is true for G_2^i , consider $([a], [x]) \in G_2^{i+1}$. By proposition 3 we need to deal with three cases.

- If for some $([b], [y]) \in G_2^i$, $([b], [y]) \preceq ([a], [x])$. Then by induction hypotheses we know $y \in \text{out}_2(G, b) = \cap\{Cn(G(V)) : b \in V, V \text{ is complete}\}$. Since $[a] \leq [b]$ and $[y] \leq [x]$ we know $a \vdash b, y \vdash x$ and $x \in Cn(y)$. Hence $x \in \cap\{Cn(G(V)) : b \in V, V \text{ is complete}\}$. Moreover, every complete set V contains a must contain b , hence $\cap\{Cn(G(V)) : b \in V, V \text{ is complete}\} \subseteq \cap\{Cn(G(V)) : a \in V, V \text{ is complete}\}$. Therefore $x \in \cap\{Cn(G(V)) : a \in V, V \text{ is complete}\}$, $x \in \text{out}_2(G, a)$.

- If there exist $([a], [y]), ([a], [z]) \in G_2^i$ such that $[x] = [y] \cdot [z]$. Then by induction hypotheses we know $y, z \in out_2(G, a) = \cap\{Cn(G(V)) : a \in V, V \text{ is complete}\}$. Therefore $y \wedge z \in \cap\{Cn(G(V)) : a \in V, V \text{ is complete}\}$ and $x \in \cap\{Cn(G(V)) : a \in V, V \text{ is complete}\}$. That is, $x \in out_2(G, a)$.
- If there exist $([b], [x]), ([c], [x]) \in G_2^i$ such that $[a] = [b] + [c]$. Then by induction hypotheses we know $x \in out_2(G, b)$ and $x \in out_2(G, c)$. Therefore $x \in \cap\{Cn(G(V)) : b \in V, V \text{ is complete}\}$ and $x \in \cap\{Cn(G(V)) : c \in V, V \text{ is complete}\}$. For every complete set V such that $b \vee c \in V$, it must be that either $b \in V$ or $c \in V$. Therefore, for every complete set V that contains $b \vee c$, $x \in Cn(V)$, which means $x \in \cap\{Cn(G(V)) : b \vee c \in V, V \text{ is complete}\}$, i.e. $x \in out_2(G, b \vee c)$, $x \in out_2(G, a)$.

3 \Rightarrow 1 : This is a special case of Theorem 1. □

Proposition 4 Given two Boolean algebra $\mathfrak{A}, \mathfrak{B}$,

1. $(\mathfrak{A}, \mathfrak{B}, A \times B)$ is a simple-minded/basic/simple-minded reusable/basic reusable Boolean joining-system.
2. If for all $i \in I$, $(\mathfrak{A}, \mathfrak{B}, S_i)$ is a simple-minded Boolean joining-system, then $(\mathfrak{A}, \mathfrak{B}, \cap_{i \in I} S_i)$ is a simple-minded Boolean joining-system. And similarly for basic/simple-minded reusable/basic reusable joining space.

Proof. Similar to the proof of Proposition 2. □

Theorem 3 For $i \in \{1, 2, 3, 4\}$, the following three propositions are equivalent:

1. $(a, x) \in deriv_i(G)$.
2. $([a], [x]) \in G_i$.
3. $x \in out_i(G, a)$.

Proof. Similar to the proof of Theorem 2. □

Theorem 6 For $i \in \{1, 2, 3\}$, the following three propositions are equivalent:

1. $(a, x) \in deriv_i^I(G)$
2. $([a], [x]) \in G_i^I$
3. $x \in out_i^I(G, a)$

Proof. Similar to the proof of Theorem 2. □

Proposition 5 Every quasi entailment \vdash_Q has the following properties:

1. $A \vdash_Q a$ for every $a \in A$.
2. If $A \vdash_Q a$ and $\{a\} \vdash_Q b$, then $A \vdash_Q b$.
3. If $A \vdash_Q a$ then $A \cup B \vdash_Q a$.

4. If $A \vdash_Q a$ and $A \vdash_Q b$, then $A \vdash_Q a \wedge b$.

Proof. Here we only prove the case of closed under conjunction. Assume $A \vdash_Q a$ and $A \vdash_Q b$, then there exist $a_1, \dots, a_m \in A$ such that $a_1 \wedge \dots \wedge a_m \vdash_q a$, $b_1, \dots, b_n \in A$ such that $b_1 \wedge \dots \wedge b_n \vdash_q b$. Therefore by monotony $a_1 \wedge \dots \wedge a_m \wedge b_1 \wedge \dots \wedge b_n \vdash_q a$ and $a_1 \wedge \dots \wedge a_m \wedge b_1 \wedge \dots \wedge b_n \vdash_q b$. Hence $a_1 \wedge \dots \wedge a_m \wedge b_1 \wedge \dots \wedge b_n \vdash_q a \wedge b$ and $A \vdash_Q a \wedge b$. \square

Theorem 7 Given an arbitrary normative system G and a set A of formulas,

$$(a, x) \in \text{deriv}_1^Q(G) \text{ iff } x \in \text{out}_1^Q(G, a)$$

Proof. (left-to-right) Assume $(a, x) \in \text{deriv}_1^Q(G)$, then we have the following cases:

1. $(a, x) \in G$. For this case, note that \vdash_Q is reflexive, hence $a \in \text{Cn}^Q(a)$ and $x \in G(\text{Cn}^Q(a))$. Again by reflexivity, $x \in \text{Cn}^Q(G(\text{Cn}^Q(a)))$.
2. (a, x) is derived by the SI^Q rule. In this case, there exist $(b, x) \in \text{deriv}_1^Q(G)$ and $a \vdash_Q b$. By the induction hypothesis we know $x \in \text{out}_1^Q(G, b) = \text{Cn}^Q(G(\text{Cn}^Q(b)))$. Now by $a \vdash_Q b$ and by the transitivity of \vdash_Q we know $\text{Cn}^Q(b) \subseteq \text{Cn}^Q(a)$. Therefore by monotony of $G(\bullet)$, which is easy to be checked, we know $G(\text{Cn}^Q(b)) \subseteq G(\text{Cn}^Q(a))$. Then by monotony of $\text{Cn}^Q(\bullet)$ we have $\text{Cn}^Q(G(\text{Cn}^Q(b))) \subseteq \text{Cn}^Q(G(\text{Cn}^Q(a)))$.
3. (a, x) is derived by the WO^Q rule. If (a, x) is derived by the WO^Q rule, then there exist $y \vdash_Q x$, $(a, y) \in \text{deriv}_1^Q(G)$. By the induction hypothesis we know $y \in \text{out}_1^Q(G, a)$, $y \in \text{Cn}^Q(G(\text{Cn}^Q(a)))$. Therefore there exist $z_1, \dots, z_n \in G(\text{Cn}^Q(a))$, $z_1 \wedge \dots \wedge z_n \vdash_Q y$. Hence $z_1 \wedge \dots \wedge z_n \vdash_Q x$ by the transitivity of \vdash_Q . Therefore $x \in \text{Cn}^Q(G(\text{Cn}^Q(a)))$.
4. (a, x) is derived by the AND rule. If (a, x) is derived by the AND rule, then there exist $(a, y) \in \text{deriv}_1^Q(G)$, $(a, z) \in \text{deriv}_1^Q(G)$ and x is $y \wedge z$. By the induction hypothesis we know $y \in \text{out}_1^Q(G, a) = \text{Cn}^Q(G(\text{Cn}^Q(a)))$ and $z \in \text{out}_1^Q(G, a) = \text{Cn}^Q(G(\text{Cn}^Q(a)))$. Recall that $\text{Cn}^Q(\bullet)$ is closed under conjunction. Then we have $y \wedge z \in \text{Cn}^Q(G(\text{Cn}^Q(a)))$.

(right-to-left) Assume $x \in \text{out}_1^Q(G, a)$, then $x \in \text{Cn}^Q(G(\text{Cn}^Q(a)))$. Therefore there exist $x_1, \dots, x_n \in G(\text{Cn}^Q(a))$ such that $x_1 \wedge \dots \wedge x_n \vdash_Q x$. From $x_1, \dots, x_n \in G(\text{Cn}^Q(a))$ we can deduce there exist $a_1, \dots, a_n \in \text{Cn}^Q(a)$ such that $(a_1, x_1), \dots, (a_n, x_n) \in G$. So we have $a \vdash_Q a_1, \dots, a \vdash_Q a_n$. Then by applying the SI_Q rule we have $(a, x_1), \dots, (a, x_n) \in \text{deriv}_1^Q(G)$. By further applying the AND rule we have $(a, x_1 \wedge \dots \wedge x_n) \in \text{deriv}_1^Q(G)$. We then derive $(a, x) \in \text{deriv}_1^Q(G)$ by using the WO_Q rule. \square

Lemma 1 $B_A^G = \bigcup_{i=0}^{\infty} B_{A,i}^G$, where $B_{A,0}^G = \text{Cn}^Q(A)$, $B_{A,i+1}^G = \text{Cn}^Q(\text{AUG}(B_{A,i}^G))$.

Proof. We first prove that $\bigcup_{i=0}^{\infty} B_{A,i}^G$ is a fixed point of f_A^G . We prove by showing the following:

1. $A \subseteq \bigcup_{i=0}^{\infty} B_{A,i}^G$: this is because $A \subseteq Cn^Q(A) = B_{A,0}^G \subseteq \bigcup_{i=0}^{\infty} B_{A,i}^G$
2. $G(\bigcup_{i=0}^{\infty} B_{A,i}^G) \subseteq \bigcup_{i=0}^{\infty} B_{A,i}^G$: For every $x \in G(\bigcup_{i=0}^{\infty} B_{A,i}^G)$, there exist k such that $x \in G(B_{A,k}^G) \subseteq B_{A,k+1}^G \subseteq \bigcup_{i=0}^{\infty} B_{A,i}^G$.
3. $Cn^Q(\bigcup_{i=0}^{\infty} B_{A,i}^G) = \bigcup_{i=0}^{\infty} B_{A,i}^G$: the right-to-left direction is obvious; for the other direction: assume $x \in Cn^Q(\bigcup_{i=0}^{\infty} B_{A,i}^G)$, then there exist $x_1, \dots, x_n \in \bigcup_{i=0}^{\infty} B_{A,i}^G$ such that $x_1 \wedge \dots \wedge x_n \vdash_Q x$. Therefore there exist k such that $x_1, \dots, x_n \in B_{A,k}^G$. Hence $x \in B_{A,k+1}^G \subseteq \bigcup_{i=0}^{\infty} B_{A,i}^G$.

With the above clauses in hand, we can prove that $f_A^G(\bigcup_{i=0}^{\infty} B_{A,i}^G) \subseteq \bigcup_{i=0}^{\infty} B_{A,i}^G$. For the other direction, we prove by induction on i that for every i , $B_{A,i}^G \subseteq f_A^G(\bigcup_{i=0}^{\infty} B_{A,i}^G)$. Here we omit the details.

So we have proved that $\bigcup_{i=0}^{\infty} B_{A,i}^G$ is a fixed point of f_A^G . To show that $\bigcup_{i=0}^{\infty} B_{A,i}^G$ is the least fixed point, we have to show that $\bigcup_{i=0}^{\infty} B_{A,i}^G$ is a subset of every fixed point of f_A^G .

Let B an arbitrary set such that $B = f_A^G(B)$. Again we prove by induction on i .

1. Base step: By monotony we know that $f_A^G(\emptyset) \subseteq f_A^G(B) = B$. That is, $B_{A,0}^G = Cn^Q(A) = Cn^Q(A \cup G(\emptyset)) \subseteq B$.
2. Inductive step: assume $B_{A,i}^G \subseteq B$. Then $f_A^G(B_{A,i}^G) \subseteq f_A^G(B) = B$. That is, $B_{A,i+1}^G = Cn^Q(A \cup G(B_{A,i}^G)) = f_A^G(B_{A,i}^G) \subseteq f_A^G(B) = B$. \square

Corollary 1. – For every $A \subseteq L$ and $G \subseteq L \times L$, $A \subseteq B_A^G$
– For every $a \in L$ and $G \subseteq L \times L$, $B_a^G = Cn^Q(B_a^G)$.
– If $x \in Cn^Q(G(B_a^G))$, then $x \in B_a^G$.

Theorem 8 Given an arbitrary normative system G ,

$$x \in Cn^Q(G(B_a^G)) \text{ iff } x \in \text{out}_3^Q(G, a)$$

Proof. (sketch) The “right to left” direction is proved by showing that $\{a\} \cup G(B_a^G) \subseteq B_a^G$ (see Corollary 1) and $Cn^Q(B_a^G) = B_a^G$ (see Corollary 1). For the other direction, notice that B_a^G is a subset of every set B such that $\{a\} \cup G(B) \subseteq B = Cn^Q(B)$. \square

Lemma 2. For every $a, b \in L, G \subseteq L \times L$, if $a \vdash_Q b$ then $B_b^G \subseteq B_a^G$.

Proof. We will prove that for every i , $B_{b,i}^G \subseteq B_{a,i}^G$. We prove by induction on i .

If $i = 0$, then $B_{b,0}^G = Cn^Q(b) \subseteq Cn^Q(a) = B_{a,0}^G$.

Assume $i = k + 1$ and $B_{b,k}^G \subseteq B_{a,k}^G$. Then $B_{b,k+1}^G = Cn^Q(\{b\} \cup G(B_{b,k}^G))$. From $B_{b,k}^G \subseteq B_{a,k}^G$ we deduce $G(B_{b,k}^G) \subseteq G(B_{a,k}^G)$. Now by the monotony of $Cn^Q(\bullet)$ we know $Cn^Q(\{b\} \cup G(B_{b,k}^G)) \subseteq Cn^Q(\{a\} \cup G(B_{a,k}^G))$. Hence $B_{b,k+1}^G \subseteq B_{a,k+1}^G$.

So we have proved for every i , $B_{b,i}^G \subseteq B_{a,i}^G$. With this result in hand, we can easily deduce that $B_b^G \subseteq B_a^G$. \square

Lemma 3. *If $x \in \text{Cn}^Q(G(B_a^G))$, then $B_a^G = B_{a \wedge x}^G$.*

Proof. By Lemma 2 we know that $B_a^G \subseteq B_{a \wedge x}^G$. For the other direction, we need to prove that for every i , $B_{a \wedge x, i}^G \subseteq B_a^G$. We prove this by induction on i .

- Base step: Let $i = 0$, we then have $B_{a \wedge x, i}^G = \text{Cn}^Q(a \wedge x)$. By Corollary 1 we have $a \in B_a^G$ and $x \in B_a^G$. Then again by Corollary 1 we have $a \wedge x \in B_a^G$.
- Inductive step: Assume for $i = k$, $B_{a \wedge x, k}^G \subseteq B_a^G$. Then $B_{a \wedge x, k+1}^G = \text{Cn}^Q(\{a \wedge x\} \cup G(B_{a \wedge x, k}^G))$. From $B_{a \wedge x, k}^G \subseteq B_a^G$ we get $G(B_{a \wedge x, k}^G) \subseteq G(B_a^G) \subseteq B_a^G$ by the monotonicity of G and by item 3 of Corollary 1. By the base step we have $a \wedge x \in B_a^G$. Then by Corollary 1 we know $\text{Cn}^Q(\{a \wedge x\} \cup G(B_{a \wedge x, k}^G)) \subseteq B_a^G$. That is, $B_{a \wedge x, k+1}^G \subseteq B_a^G$. \square

Theorem 9 If $(a, x) \in \text{deriv}_3^Q(G)$ then $x \in \text{Cn}^Q(G(B_a^G))$

Proof. Assume $(a, x) \in \text{deriv}_3^Q(G)$, then we prove by induction on the length of the derivation.

- (Base step) Assume $(a, x) \in G$, then by Corollary 1 we have $a \in B_a^G$. Hence $x \in G(B_a^G) \subseteq \text{Cn}^Q(G(B_a^G))$.
- Assume $(a, x) \in \text{deriv}_3^Q(G)$ and it is derived by using SI^Q from $(b, x) \in \text{deriv}_3^Q(G)$ and $a \vdash_Q b$. Then by inductive hypothesis we have $x \in \text{Cn}^Q(G(B_b^G))$. By Lemma 2 we know $B_b^G \subseteq B_a^G$. Therefore we further have $G(B_b^G) \subseteq G(B_a^G)$, $\text{Cn}^Q(G(B_b^G)) \subseteq \text{Cn}^Q(G(B_a^G))$. Hence $x \in \text{Cn}^Q(G(B_a^G))$.
- Assume (a, x) is $(a, x_1 \wedge x_2)$, $(a, x) \in \text{deriv}_3^Q(G)$ and it is derived by using AND from (a, x_1) and (a, x_2) . Then by inductive hypothesis we have $x_1, x_2 \in \text{Cn}^Q(G(B_a^G))$. Therefore $x_1 \wedge x_2 \in \text{Cn}^Q(G(B_a^G))$.
- Assume $(a, x) \in \text{deriv}_3^Q(G)$ and it is derived by using WO^Q from $(a, x_1) \in \text{deriv}_3^Q(G)$ and $x_1 \vdash_Q x$. Then by inductive hypothesis we have $x_1 \in \text{Cn}^Q(G(B_a^G))$. Since $x_1 \vdash_Q x$, we can prove that $x \in \text{Cn}^Q(G(B_a^G))$.
- Assume $(a, x) \in \text{deriv}_3^Q(G)$ and it is derived by using CT form $(a, x_1) \in \text{deriv}_3^Q(G)$ and $(a \wedge x_1, x) \in \text{deriv}_3^Q(G)$. Then by inductive hypothesis we have $x_1 \in \text{Cn}^Q(G(B_a^G))$ and $x \in \text{Cn}^Q(G(B_{a \wedge x_1}^G))$. Then by Lemma 3 we have $B_a^G = B_{a \wedge x_1}^G$. Therefore $x \in \text{Cn}^Q(G(B_a^G))$. \square

Lemma 4. *For all i , if $b \in B_{a, i}^G$ and $(b, x) \in G$, then $(a, x) \in \text{deriv}_3^Q(G)$*

Proof. We prove by induction on i .

- Base step: Let $i = 0$. Then $b \in B_{a, 0}^G = \text{Cn}^Q(a)$. Hence $a \vdash_Q b$. Therefore we can apply SI^Q to $a \vdash_Q b$ and $(b, x) \in G$ to derive $(a, x) \in \text{deriv}_3^Q(G)$.
- Inductive step: Assume for $i = k$, if $b \in B_{a, k}^G$ and $(b, x) \in G$, then $(a, x) \in \text{deriv}_3^Q(G)$. Now let $b \in B_{a, k+1}^G$. Then $b \in \text{Cn}^Q(\{a\} \cup G(B_{a, k}^G))$, and there exist $b_1 \dots b_n \in G(B_{a, k}^G)$ such that $a \wedge b_1 \wedge \dots \wedge b_n \vdash_Q b$. Then apply SI^Q to $(b, x) \in G$ and $a \wedge b_1 \wedge \dots \wedge b_n \vdash_Q b$ we have $(a \wedge b_1 \wedge \dots \wedge b_n, x) \in \text{deriv}_3^Q(G)$.

Note that for each $i \in \{1, \dots, n\}$, from $b_i \in G(B_{a,k}^G)$ we know there is $a_i \in B_{a,k}^G$ such that $(a_i, b_i) \in G$. Now by inductive hypothesis we have $(a, b_i) \in \text{deriv}_3^Q(G)$. Then applying the AND rule we have $(a, b_1 \wedge \dots \wedge b_n) \in \text{deriv}_3^Q(G)$. From $(a, b_1 \wedge \dots \wedge b_n) \in \text{deriv}_3^Q(G)$ and $(a \wedge b_1 \wedge \dots \wedge b_n, x) \in \text{deriv}_3^Q(G)$ we adopt the CT rule to derive $(a, x) \in \text{deriv}_3^Q(G)$. \square

Theorem 10 If $x \in \text{Cn}^Q(G(B_a^G))$, then $(a, x) \in \text{deriv}_3^Q(G)$.

Proof. Assume $x \in \text{Cn}^Q(G(B_a^G))$, then there exist $x_1, \dots, x_n \in G(B_a^G)$ such that $x_1 \wedge \dots \wedge x_n \vdash_Q x$. For each $i \in \{1, \dots, n\}$, from $x_i \in G(B_a^G)$ we know there is $a_i \in B_a^G$ such that $(a_i, x_i) \in G$. From $a_i \in B_a^G$ we know there exist k such that $a_i \in B_{a,k}^G$. Now by Lemma 4 we know $(a, x_i) \in \text{deriv}_3^Q(G)$. Then applying the AND rule we have $(a, x_1 \wedge \dots \wedge x_n) \in \text{deriv}_3^Q(G)$. Then by the WO^Q rule we have $(a, x) \in \text{deriv}_3^Q(G)$. \square

Theorem 11 The following three propositions are equivalent: for $i \in \{1, 3\}$

1. $(a, x) \in \text{deriv}_i^Q(G)$
2. $([a], [x]) \in G_i^Q$
3. $x \in \text{out}_i^Q(G, a)$

Proof. Similar to the proof of Theorem 2. \square

Observation 1 For every joining-system $\mathbb{S} = (A, B, S)$, if S is finite then $\text{core}(\mathbb{S}) \neq \emptyset$.

Proof. The proof is trivial. Due to the fact that S is finite, there is no infinite descending chain on \prec . \square

Observation 2 For all joining-system $\mathbb{S} = (\mathfrak{A}, \mathfrak{B}, S)$, if S is finite, then for any $(a, x) \in S$, there exists $(b, y) \in \text{core}(\mathbb{S})$ such that $(b, y) \preceq (a, x)$.

Proof. Let (a, x) be an arbitrary norm in S . If $(a, x) \in \text{core}(\mathbb{S})$, then $(a, x) \preceq (a, x)$ and we are done. If $(a, x) \notin \text{core}(\mathbb{S})$, then (a, x) is not a minimal norm. Hence there exist some (b, y) such that $(b, y) \prec (a, x)$. If $(b, y) \in \text{core}(\mathbb{S})$ then we are done. If not, then there exist some (c, z) such that $(c, z) \prec (b, y)$. Since S is finite, this procedure will stop at some point. Then by transitivity of \preceq , there must exist some $(a', x') \in \text{core}(\mathbb{S})$ such that $(a', x') \preceq (a, x)$. \square

Observation 3 For any joining-system $\mathbb{S} = (\mathfrak{A}, \mathfrak{B}, S)$ and $\mathbb{S}' = (\mathfrak{A}, \mathfrak{B}, S')$, if both S and S' are finite, then $\text{core}(\mathbb{S}) = \text{core}(\mathbb{S}')$ iff $S = S'$. \square

Proof. The right to left direction is trivial. For the left to right direction. Assume $\text{core}(\mathbb{S}) = \text{core}(\mathbb{S}')$. For any $(a, x) \in S$, by Observation 2 there exist $(b, y) \in \text{core}(\mathbb{S})$ such that $(b, y) \preceq (a, x)$. By assumption we know $(b, y) \in \text{core}(\mathbb{S}')$. Then by the definition of joining space we know $(a, x) \in S'$. Therefore $S \subseteq S'$. Similarly we can prove $S \supseteq S'$. \square